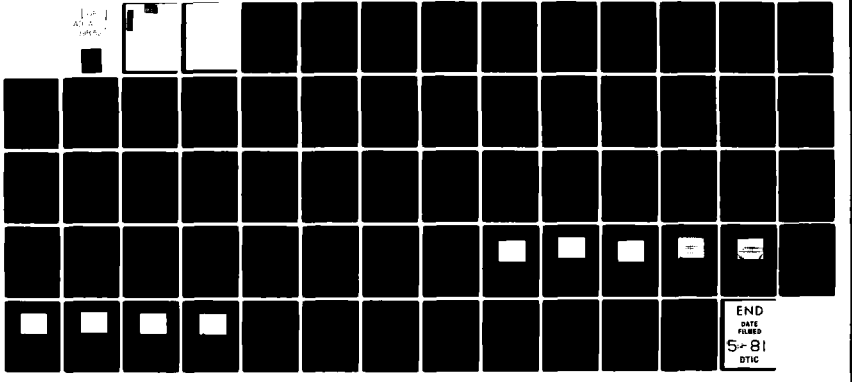
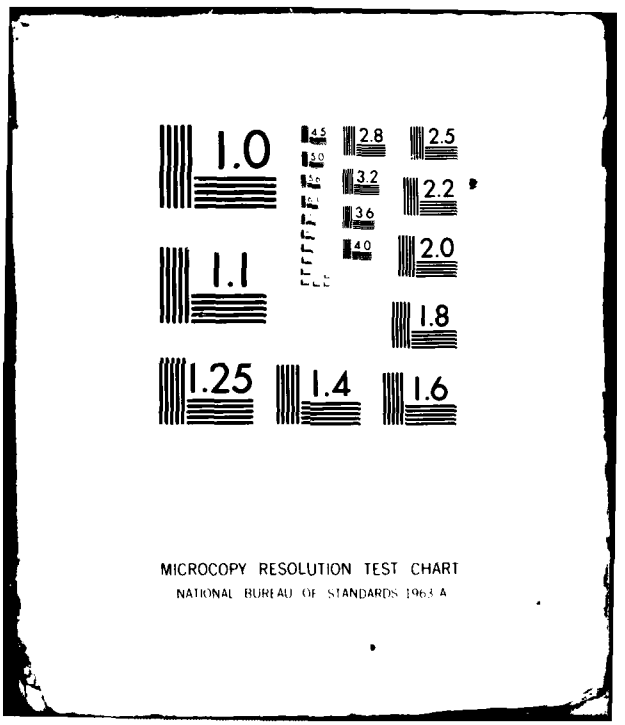
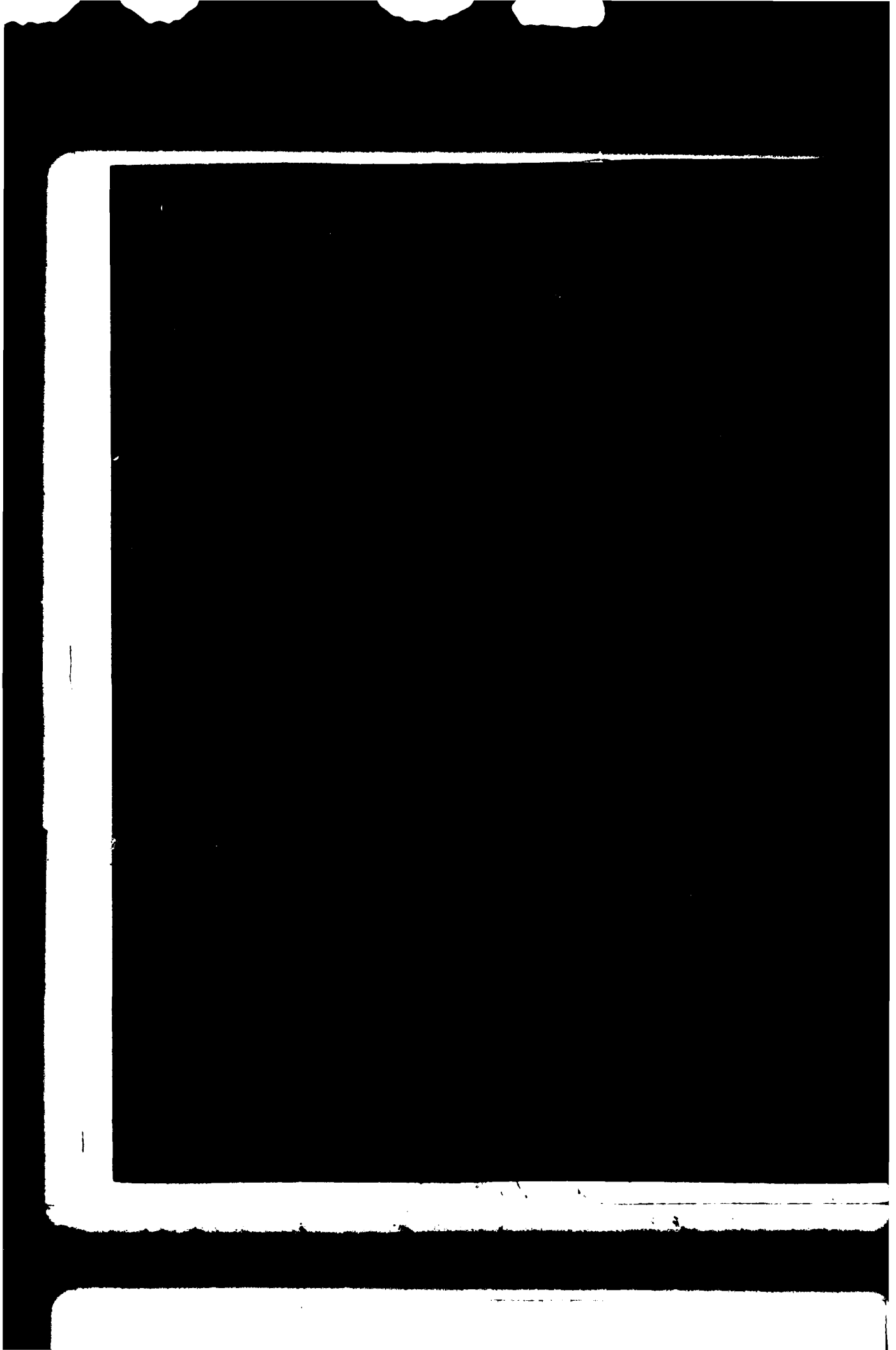
AD-A098 521 MASSACHUSETTS INST OF TECH LEXINGTON LINCOLN LAB F/G 17/9
THE DIGITAL SIGNAL PROCESSOR FOR THE ALCOR MILLIMETER WAVE RADA--ETC(U)
NOV 80 R A FORD F19628-80-C-0002
UNCLASSIFIED TN-1980-51 ESD-TR-80-221 NL




1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A



MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LINCOLN LABORATORY

THE DIGITAL SIGNAL PROCESSOR FOR THE ALCOR MILLIMETER WAVE RADAR

R. A. FORD
Group 91

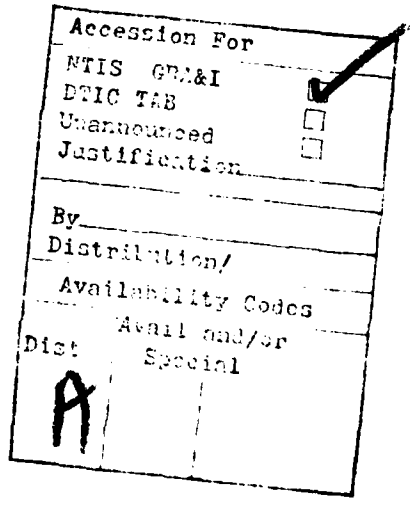
TECHNICAL NOTE 1980-51

6 NOVEMBER 1980

LEXINGTON MASSACHUSETTS

ABSTRACT

This report describes the use of an array processor for real time radar signal processing. Pulse compression, range marking, and monopulse error computation are some of the functions that will be performed in the array processor for the millimeter wave ALCOR radar augmentation. Real time software design, processor architecture, and system interfaces are discussed in the report.



CONTENTS

1. INTRODUCTION

The Digital Signal Processor (DSP) implements the functions of pulse compression, range marking, and monopulse channel processing in the millimeter wave augmentation to the ALCOR radar. As an instrumentation radar, it is desirable to have the flexibility to handle multiple waveforms and to be able to change or add new waveforms with relative ease. A programmable digital signal processor has the flexibility required for this application but has limitations in its speed capabilities. Based on experience with the Lincoln Laboratory LRIR radar, it was decided that the speed capability is adequate for real time processing requirements of the millimeter wave radar.

There are, today, several programmable signal processors readily available on the commercial market that will perform a 1024 point complex FFT in under 5 milliseconds. The model chosen was the AP-120B made by Floating Point Systems, Inc. A block diagram of the unit is shown in Figure 1. It is a TTL technology device utilizing floating point arithmetic with a single real multiplier and a single real adder, both pipelined. The instruction rate is 6 MHz and the maximum computation rate is 12 million floating point operations per second, although this is a peak rate which is rarely attained. The program instruction word is 64 bits wide and, like all the pipelined array processors, presents a rather formidable programming task, several times more difficult than assembly language

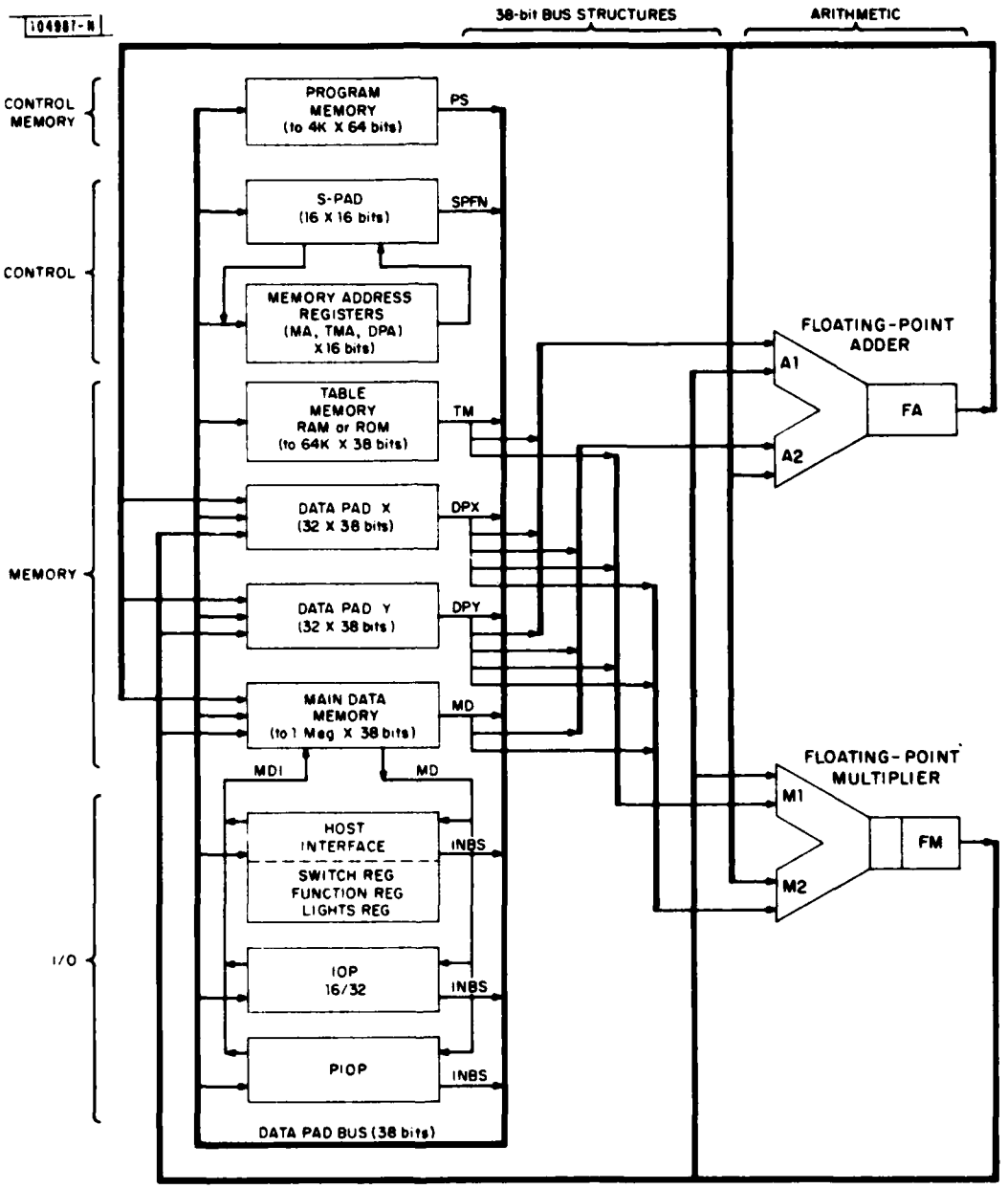


Fig. 1. AP-120B block diagram.

on a general purpose computer, for example. The AP-120B is synchronous rather than the much more common asynchronous architecture. In use, this difference manifests itself as a reduction in programming effort by as much as a factor of two. This is an important factor in the successful integration of an array processor into the complete system. As purchased, the DSP has 4 K words of program memory, 16 K words of main data memory, 2 K words of supplementary random access memory, and a programmable input/output processor. It occupies about half a rack of physical space.

2. SOFTWARE OVERVIEW

In developing the real time software for the array processor, care was taken to follow the rules for structured programming as would be done in any high level language on a general purpose computer, rather than use the magic words "real time" as an excuse to ignore the rules of good programming practice. The software for the DSP is highly modular consisting of over 50 separate subroutines. The main advantages to the user are the ease of program validation and the ease with which modifications or additions can be made. While the radar currently utilizes only a basic linear frequency modulation (chirp) pulse, the software has been designed for the inclusion of other waveforms requiring other processing techniques or for pulse to pulse processing, either coherent or noncoherent. As another example, the subroutine used for range marking is one that can be easily revised as new algorithms are developed

and tested for use against various types of targets.

It is important that the emphasis for speed be placed where the payoff is greatest, that is, in the coding of the vector subroutines, or, in different terminology, the microcode. It is here that two-thirds of the processing time will be spent executing only five percent of the program instructions.

In addition to whatever programming languages are in use in the host computer, the DSP has its own assembly language which was used for all internal programming. The I/O processor for the AP-120B is also a programmable device and has its own unique programming language. Both of these will contain microcode which, unlike normal assembly code for general purpose computers, may contain up to five or six completely distinct instructions that are executing simultaneously.

The complete program listing contains over 5000 lines of source code, not including the library subroutines supplied by the manufacturer. When resident in the AP-120B, it will occupy just over half of the available program memory.

3. HARDWARE INTERFACE OVERVIEW

The radar coherent video is sampled by eight 5 MHz A/D converters in four channels: PP, Az, El, and OP. As this rate is substantially higher than could be accepted directly by the I/O processor, it is buffered by a

track data buffer consisting of four hardware memories each holding up to 512 complex radar samples which may be read independently. In addition, there is an aux buffer, 16 words* long, whose contents are described in Table 1. The first two words contain important control information for the DSP and are described in Table 2. The radar itself will operate up to a 2000 Hz PRF but the DSP is limited to a nominal 200 Hz processing rate. Unless otherwise indicated, PRI means the effective PRI into the DSP. The track data buffer and aux buffer are loaded only on those PRI's which the DSP will process. The aux buffer, or at least the first DSP control word is read to determine the basic processing control parameters on each PRI.

A display is also connected to the DSP which normally serves as the radar A-scope. As the pulse compression is performed in the DSP, this is the only point in the system where the compressed pulse can be seen. The output data is in floating point format to facilitate easy conversion to a logarithmic scale. The display is restricted to either 256 or 512 points.

4. EXECUTIVE LEVEL SOFTWARE

The software in the DSP may be functionally divided into two categories: the executive level software and the signal processing software. The

* At the time this report was written, the aux buffer was 16 words long but with the understanding that it was probably to be increased to 32 words. As the exact length is not important to the concepts presented here, 16 words has been assumed throughout with the understanding that the final configuration may very well contain 32 words.

TABLE 1

AUX BUFFER DESCRIPTION

Word	Contents
0	DSP control
1	DSP control
2	Gain control, attenuation
3	Azimuth encoder Word 1
4	Azimuth encoder Word 2
5	Elevation encoder Word 1
6	Elevation encoder Word 2
7	PRI (LSB = 1.6 μsec)
8	Ambiguity count (lower 6 bits only)
9	Ambiguous receive time (LSB = 1.6 μsec)
10-15	To be determined or spares

TABLE 2

DSP CONTROL WORDS IN THE AUX BUFFER

Word 0: Digital Signal Processor Control

Bit 0-2	Processing Mode. Taken as an integer with bit 2 the LSB:	
	Integer	Mode
	0	Test
	1-2	Spares
	3	Real time mission mode
	4	Spare
	5	Calibration
	6	Spare
	7	High sensitivity acquisition
3	0 = Inhibit angle channel transfer to DSP	
	1 = Enable angle channel transfer to DSP	
4	0 = Inhibit OP channel transfer to DSP	
	1 = Enable OP channel transfer to DSP	
5	Spare	
6-15	Number of complex radar samples to be read from the track data buffer in each channel to be processed. Bit 15's the LSB.	

TABLE 2 (Cont'd)

Word 1: Digital Signal Processor Control

Bit 0	0 = Receive narrowband pulse		
	1 = Receive wideband pulse		
1	0 = Do not double receive bandwidth		
	1 = Double receive bandwidth		
	Together, bits 0 and 1 mean		
	Bit 0	Bit 1	Receive Bandwidth
	0	0	6 MHz
	0	1	12 MHz
	1	0	500 MHz
	1	1	1000 MHz
2	0 = 35 GHz receive frequency		
	1 = 95 GHz receive frequency		
3	1 = Enable transfer from DSP to computer		
	0 = Inhibit transfer from DSP to computer		
4-5	Spares		
6	0 = 256 complex point FFT		
	1 = 512 complex point FFT		
7-8	Spares		

TABLE 2 (Cont'd)

9-11 Track mode

Bit			Mode
9	10	11	
0	0	0	Centroid
0	0	1	Leading edge
0	1	0	Trailing edge
0	1	1	Adaptive threshold
1	0	0	Spare
1	0	1	"
1	1	0	"
1	1	1	"

12 0 = ungated range window

1 = gated range window

13-15 Weighting selection

Bit			Weighting
13	14	15	
0	0	0	Hamming
0	0	1	Uniform weighting
0	1	0	35 GHz, 500 MHz PP
0	1	1	35 GHz, 1000 MHz PP
1	0	0	95 GHz, 500 MHz PP
1	0	1	95 GHz, 1000 MHz PP
1	1	0	Spare
1	1	1	Spare

executive software acts as a form of "traffic cop" between the raw data coming from the radar itself, the results of the processing going to the host computer, the intrapulse processing which must be completed within a single PRI, and the interpulse processing which processes data from multiple radar PRI's.

Figure 2 is a block diagram of the executive software in the signal processor. This program waits for the end of the data transfer from the A/D converters, initiates the intrapulse processing and, if the last set of data in an integration sequence has been received, starts the interpulse processing. It must also keep track of the idle time, i.e., the percent of real time that is not being used by the real time program.

The I/O processor is reset for the next transfer immediately upon completion of the last transfer. Because of this and the fact that the data is double buffered in the DSP, the time required for the reset (under 40 μsec) is of no consequence in determining the maximum speed capabilities of the DSP.

Two things are done before the actual initiation of the intrapulse processing. At this point an overload caused by too much interpulse processing can easily be detected and this is done. Also, this is the point where the multimode capability of the DSP is enabled. One of the parameters available from the aux buffer on each PRI is a mode indicator which determines the type of processing that should be performed on the radar

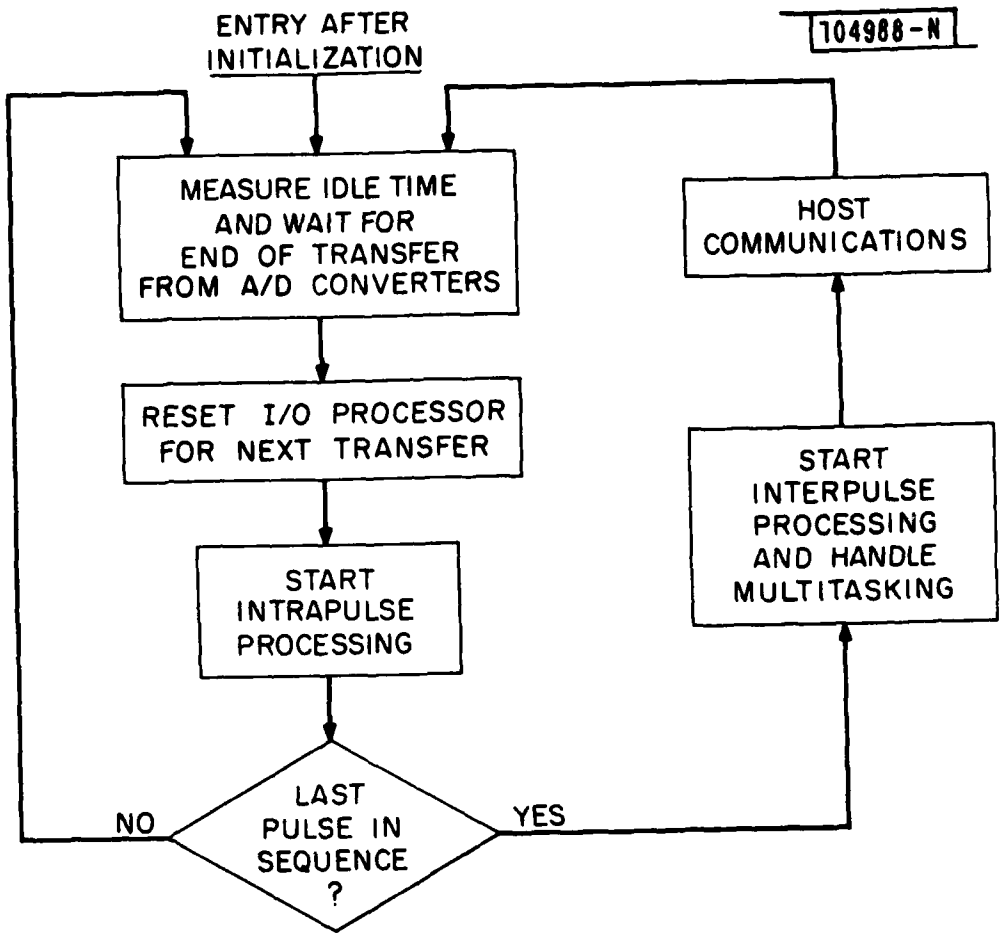


Fig. 2. Executive software block diagram.

video during that PRI. Eight independent processing modes are available with four currently allocated. The correct intrapulse processing mode for each PRI is then initiated.

The intrapulse processing nominally must be completed before the end of the next data transfer (one PRI later). This is repeated until N sets of data have been received, i.e., N intrapulse processing steps have been completed. The results of this processing serve as the input to the interpulse processing. The interpulse processing runs at a lower priority than the intrapulse processing. This situation is depicted in Figure 3. In this example, four data sets are processed (intrapulse) before the interpulse processing starts. Note that the interpulse processing for block k is being interrupted by the intrapulse processing for data set $k + 1$. Any subroutine in the interpulse processing sequence which is executing at the time the data transfer ends may run to completion. This is an important point in the design philosophy of the software and comes about primarily because of restrictions imposed by the architecture of the signal processor. This is the fact that the AP-120B itself is not interruptible (although its I/O processor is), and the arithmetic pipelines have no hardware provisions for saving the contents of the clocking registers. This latter fault is a common characteristic of this generation of array processors. To steal the terminology from Control Data, an Exchange Jump instruction is needed which will save the contents of all the clocking registers in the arithmetic and memory pipelines.

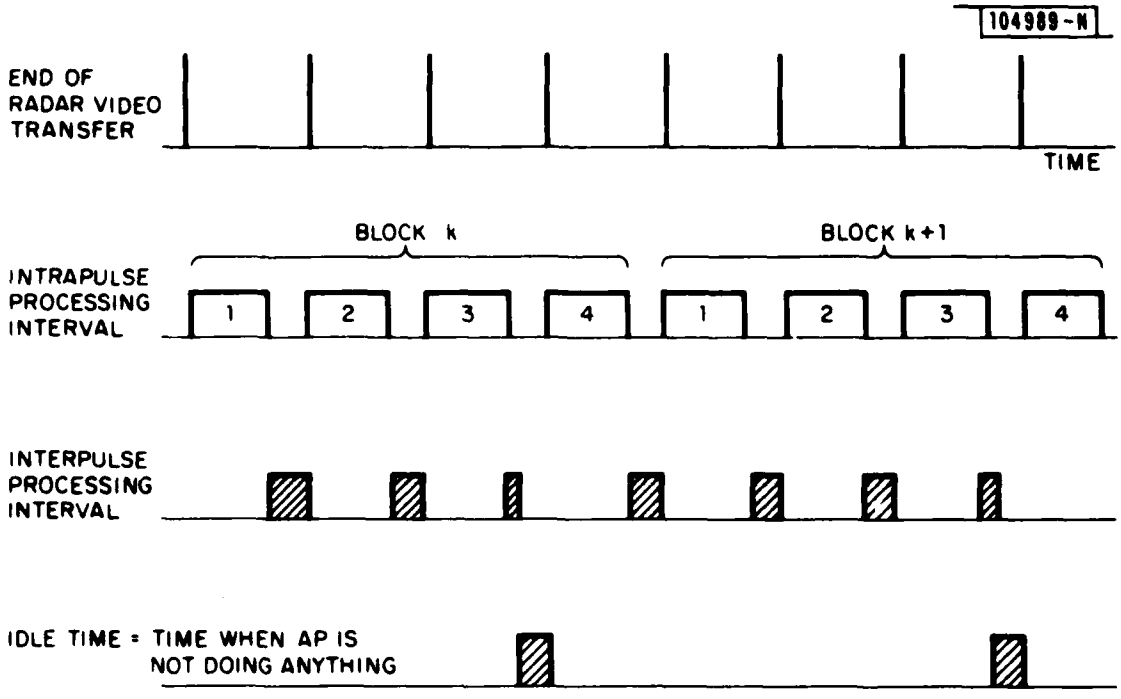


Fig. 3. Timing sequence.

As with the intrapulse processing, the same multimode capability is available in the interpulse processing. After every significant (measured in units of execution time) subroutine call in the interpulse processing sequence, a routine to handle the multitasking is called. In this, a test is made to see if the transfer from the A/D converters has been completed, and if so, the I/O processor is reset and the intrapulse processing for the current PRI is initiated. At the end of the interpulse processing, the results are transferred to host memory by the DSP and the host is interrupted.

5. SIGNAL PROCESSING SOFTWARE

Four of the eight possible processing modes have already been allocated with the rest currently being spares. One of these is simply a test mode in which the DSP does not process the radar data but merely transfers it to the host computer. Another is a calibration mode where the pulse compression is performed in all four channels and the coherent output signal is transferred to the host computer. Target tracking is not performed by the DSP in this mode. The third, and most important and complex, is the prime mission mode. Last is a mode which has been reserved for a high sensitivity acquisition application in which signals at the full radar PRF of 2000 Hz would be coherently processed by the DSP yielding an enhancement of up to 20 dB in the radar system sensitivity. Each of these modes is selectable in real time on a pulse by pulse basis and may be operated continuously or interleaved with the standard mission mode processing on a

pulse by pulse basis as desired by the user.

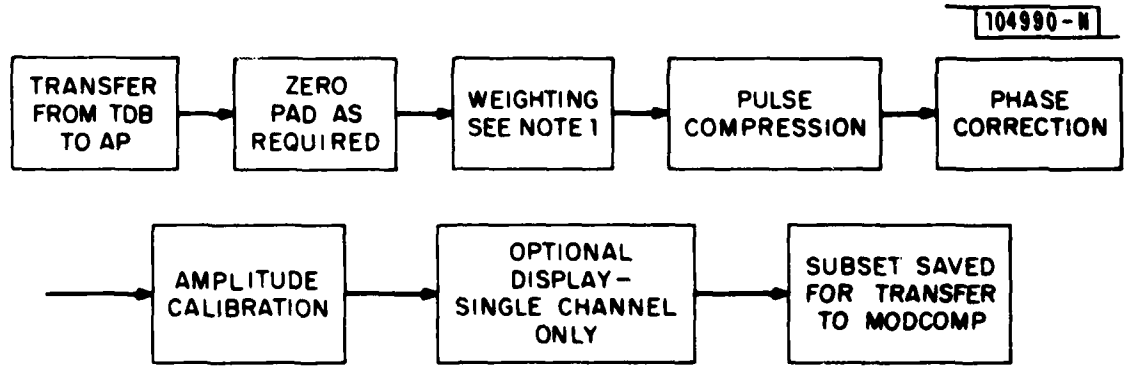
5.1 Test Mode

The radar samples in all four channels of the track data buffer are transferred to the AP's main data memory converting from 2's complement integers to floating point numbers on the fly. No other processing of the data is performed in this mode. A subset of the radar video, independently selectable in each channel, is concatenated and transferred to the host computer. The selection parameters are loaded into the AP at initialization time as part of a fixed point parameter buffer. This subset may range from zero to the entire track data buffer.

5.2 Calibration Mode

The basic processing in this mode is the pulse compression of all four radar channels. The amplitude signal across the range window from any one of these channels may be displayed either to assist in receiver alignment or just as a general indication that all is well. Figure 4 depicts the general flow of the processing. The DSP does not perform any tracking in this mode. Amplitude detection and tracking will be performed by the host computer if required. As this is a calibration mode, no attempt has been made to minimize the execution time which is about 10 milliseconds.

All four channels are processed identically, the only exception being the use of OP transversal equalizers in the OP channel in place of the PP



NOTE 1: SELECT FROM HAMMING, NONE, 35/500, 35/1000, 95/500, 95/1000

Fig. 4. Calibration mode processing.

transversal equalizers which may be selected for the PP, Az, and El channels. Four TEs may be selected for sidelobe reduction in each polarization, corresponding to the 500 MHz or 1000 MHz bandwidths at either the 35 GHz or 95 GHz center frequencies.* In addition to the TEs, either no weighting at all or a simple Hamming amplitude weighting may be selected. The sidelobe suppression in this radar is implemented as a complex array multiply of the input signal with a reference function prior to the pulse compression, which itself is simply a Fast Fourier Transform. The weighting tables are loaded into memory at initialization time. The determination of the contents of these tables is not within the purview of this report. But briefly, it is well known from the theory of linear systems that convolution in the time domain is equivalent to multiplication in the frequency domain, or in our context, a sidelobe suppression filter placed after the FFT is equivalent to a complex multiply prior to the FFT. Since the latter is computationally more efficient, this is the approach taken. By measuring the impulse response of the radar, using a sphere target for example, an ideal pulse shape and a measured pulse shape can be determined. The weighting function is then

$$\frac{\text{FFT}^{-1}\ \{\text{ideal pulse shape}\}}{\text{FFT}^{-1}\ \{\text{measured pulse shape}\}}$$

* Transversal equalizers should only be used on the wideband waveforms. Although nothing in the code prevents their being used in narrowband, it makes no sense from a signal processing point of view.

except that the computation should be performed recursively approaching the ideal performance slowly. The reasons for this can be found in the theory of signals which are constrained in both duration and bandwidth.

If the number of radar samples read from the track data buffer is not a power of two, it is necessary to pad the input array with zeros prior to the FFT. This is also the means for interpolating the output pulse shape if desired.

To insure no phase change across the pulse, a phase correction is placed on the FFT output which is of the form:

$$\exp\left\{j\,\pi\,\frac{N-1}{M}\,k\right\}$$

N = number of radar samples

M = size of FFT

The output array is converted to represent range in an up chirp radar, i.e., positive frequencies in the first half of the array and negative frequencies in the second half, with the center of the range window, which is the d.c. term out of the FFT, occurring in bin number N/2-1, starting from zero. The amplitude calibration compensates for the number of radar samples and the amount of zero-padding in the FFT and insures that an input signal of the form:

$$A\,e^{j\omega t}$$

will have an output power of k $|A|^2$ where k is the channel calibration constant. This is determined by two factors, one representing the differences among the four channels and the other being waveform dependent. This latter is determined by bits 0-2 of word 1 in the track aux buffer. The calibration constant is given by

$$\frac{(4\pi)^3 \, L \, (\text{A/D volts per count})^2}{P_T \, \lambda^2 \, G^2_{ant} \, (\text{net system power gain}) \, R_\Omega}$$

where:

L = system losses

P_T = peak transmitter power

G_{ant} = antenna gain

R_Ω = input impedance of A/D converter

If the calibration constant is not used, i.e., set to unity, the output (voltage) from the DSP is in units of A/D counts. As defined above, the output (power) is target cross section but uncompensated for AGC or target range. An option is available to use the range available in the aux buffer as part of the calibration thereby yielding target cross section in square meters, except for the AGC.

In an identical manner to the test mode, a subset of the coherent signal in each channel is saved for transfer to the host computer. This subset is independently selectable for each channel and may range from zero to the entire range window.

5.3 Mission Mode

This is the primary radar tracking mode which performs the basic operations of pulse compression, range marking, and monopulse channel processing as shown in Figure 5. In this mode, only the PP channel is compressed over the entire range window. The weighting, zero padding, and pulse compression is essentially identical to the calibration mode. As in that mode, the output of the FFT is converted to range for an up chirp radar and the received signal power of the compressed waveform is then displayed in an A-scope format.

The target detection, or range marking, algorithm is dynamically selectable in real time by the low order three bits of word one of the aux buffer. Four of the eight possible choices have already been allocated and the software has been expressly designed to enable new algorithms to be added or the current ones altered. This is an important design philosophy for a measurements radar for the range marking algorithm is the single one most likely to change as experience is gained against varying target complexes or other factors which do not yield a point source target. Three of the

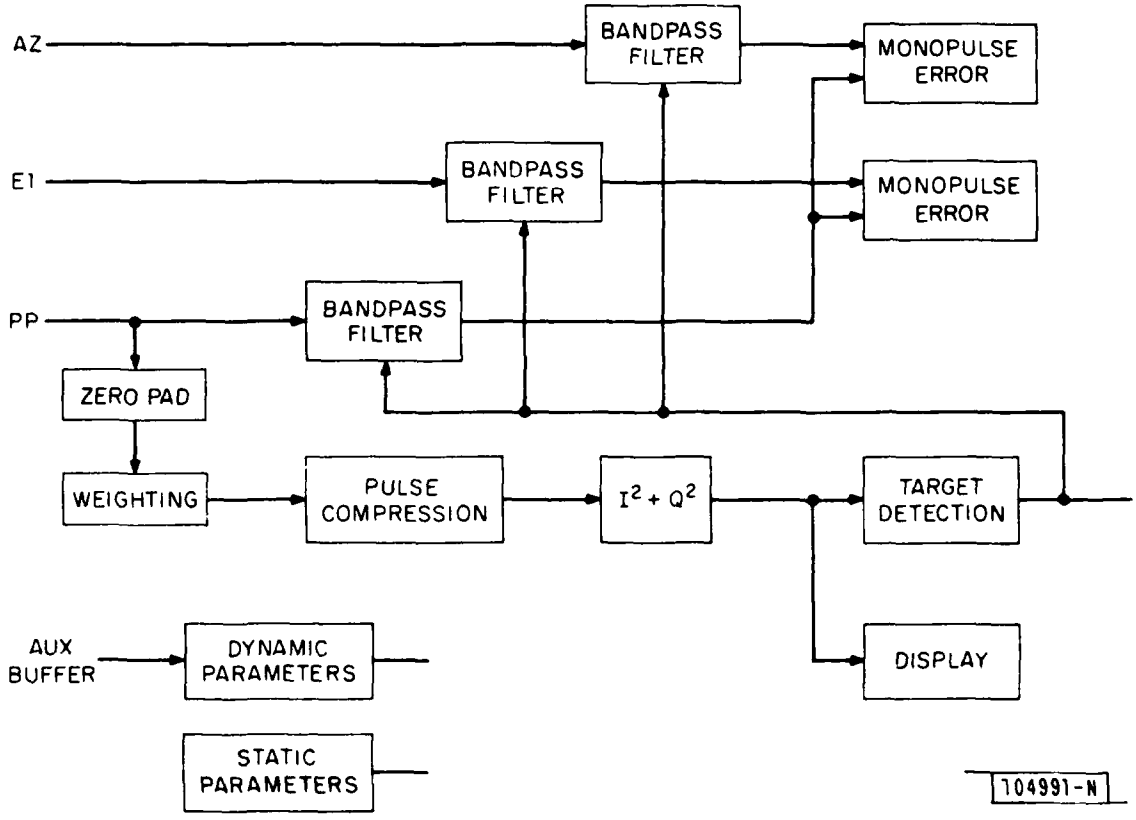


Fig. 5. Mission mode processing.

present choices emulate the existing ALCOR C-band range markers while the fourth is an adaptive threshold detector. The range marking routine may be applied to the entire range window or only to a gated subset. The choice is a dynamic parameter in word one of the aux buffer while the gating parameters themselves, i.e., position and width, are set at initialization time independently for each of the eight range marking routines. These parameters are relative to the 256 point FFT. If the 512 point FFT is selected , the midpoint of the gate but not the gatewidth is doubled. This will keep the gate in the same position on the display.

To facilitate the introduction of new routines in the future, each range marking routine has available to it two special fixed point parameters and two special floating parameters which are set at initialization time. These are in addition to the expected parameters such as number of range bins, starting address of the array, location of output answers, etc.

The first detector is a simple peak detector which scans the input array for the maximum value above a noise threshold. A fixed threshold is used which is determined prior to the mission and is loaded as the first special floating point parameter. If a peak is found above the noise threshold, and if it does not lie on either extremity of the input array, a quadratic interpolation is performed on the three samples centered at the peak. That is, the range error will be improved by adding the fractional bin number:

$$\frac{1}{2}\frac{A_{i-1} - A_{i+1}}{A_{i-1} - 2A_i + A_{i+1}}$$

If the peak does lie on the gate boundary, then the interpolation is not performed. With an ungated range window this routine would be used for long range target acquisition; while, with a very selective gate width, down to a minimum of three range bins, it could be used for tracking in late re-entry where the narrow gate would make it insensitive to debris passing through the range window.

The leading edge detector is next. In this detector the entire range window, irrespective of gating, is scanned to find the peak. This is assumed to be the peak wake cross-section and it is multiplied by a factor, less than unity, which represents the residual sidelobe level in the range window. The detection threshold is then computed as this or the receiver noise level, whichever is greater. The sidelobe level is the second special floating point parameter and, as with the peak detector, the receiver noise is the first special floating point parameter. With the detection threshold established, the range signal is scanned from near range to far range to find the first threshold crossing. If successful, this establishes the edge of a new gate, nominally three bins wide, although the actual width is to be found in the first special fixed point parameter. The range error for this new subgate is then computed as if the peak detector were in use. Figure 6 depicts the operation of this detector. The third detector is a

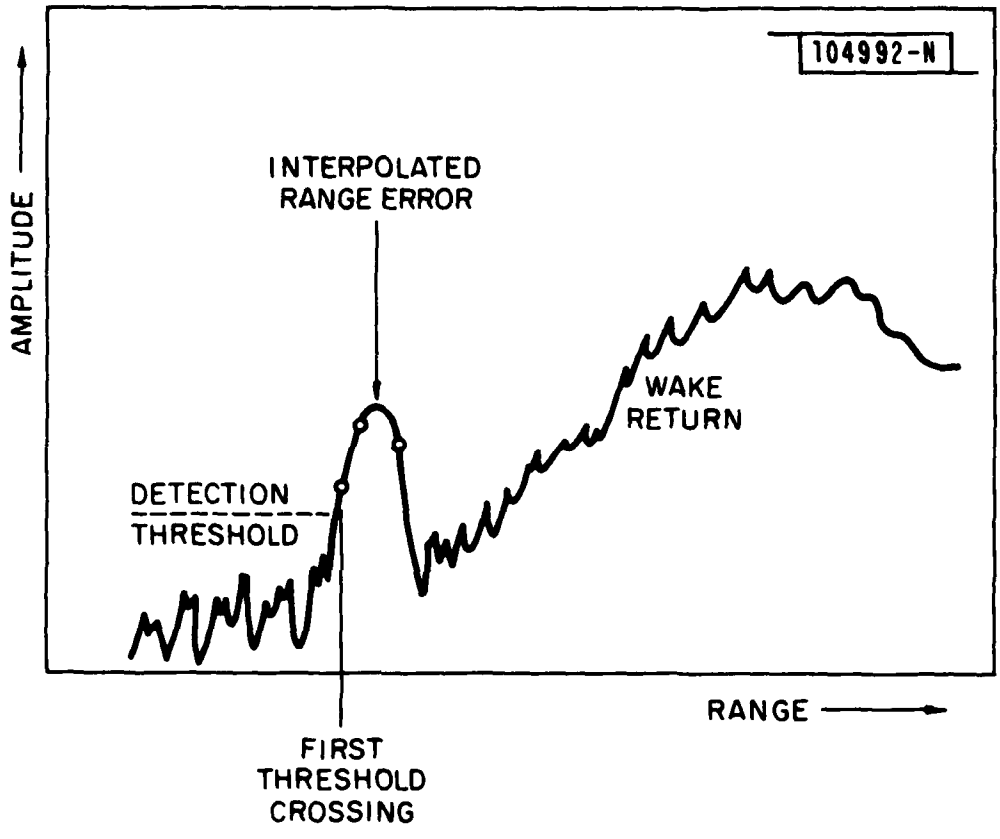


Fig. 6. Leading edge detector operation.

trailing edge version of this whereby the range signal is scanned from far range to near range. Otherwise, the operation is essentially identical.

The adaptive threshold detector does not use a fixed noise threshold but, as the name implies, measures the noise adaptively in the range window and adjusts the threshold accordingly. For each range bin which is being tested for a target the assumed noise in a gate on each side, but excluding the bin under test, is averaged and the threshold set at the appropriate signal to local noise level. This detector will select whichever potential target in the range window exceeds its local noise by the greatest margin. The noise gate width, on each side, is in the first special fixed point parameter while the minimum signal-to-noise ratio for a detection is given by the first special floating point parameter. As with the other routines, the target position is quadratically interpolated for a better range measurement.

This is computationally the slowest of the routines, requiring 1 μsec per array element. The peak detector is the fastest at 1/3 μsec while the edge detectors have an average time of 1/2 μsec. These numbers are actually very significant for all array processors are notoriously inept at performing data dependent operations. Without getting into programming details or counting instruction cycles, suffice it to say that it is a tribute to the architecture of this array processor that all of these data dependent sub-routines have been relatively easy to program and operate at surprisingly high efficiency.

The range calibration is determined by bits 0 and 1 of word 1 in the aux buffer. The calibration factor is part of the initialization parameters and is the size of the entire range window, given by

$$(c/2)\ (f_s\ T/B)$$

where

c = speed of light

f_s = A/D sampling frequency

T = pulse length

B = bandwidth

If bit 3 of word 0 in the aux buffer is set, the azimuth and elevation error channels, as well as a reference sum channel are now compressed in only a single range bin using a bandpass filter. By using the interpolated range position there will be no loss in sensitivity since the compressed pulse position is steered to the exact target location. This is an important point in the software design for the monopulse error is not simply computed at the nearest range sample, which would be the easiest thing to do, but is computed at the exact interpolated range mark point where the signal to noise ratio is greatest.

The FFT, for example, builds its bandpass filters by computing

$$\Sigma\ S_i\ e^{j\phi ik/N}$$

for integer values of k. In the monopulse computation k has no such constraint. The penalty for this is minor, under 200 μsec, being the time required to redundantly compress the sum channel. The routine that implements the bandpass filter is unique, being the only one that actually attains the full 12 Mflops computation rate. To further minimize the loss in sensitivity, weighting is not normally applied in these channels. However, because the range sidelobes of a large wake may mask the target if left unweighted, the option for Hamming weighting has been included.

The monopulse error signal in each channel is computed as

$$\mathrm{Re}\left\{\frac{\Delta}{\Sigma}\ k\right\}$$

where k is a complex calibration constant, one for each channel. The phase angle of k is used to adjust for the phase imbalance of the sum and difference channels while the magnitude calibrates the error signal into radians. Although not shown in Figure 5, the OP channel may be processed along with the angle error channels by setting bit 4 in word 0 of the aux buffer. This will make available the OP cross section of the target detected in the PP channel. As OP angle channel data is not available in the ALCOR millimeter wave radar, one cannot angle track in OP. Also not shown, is the measurement of the noise in the range window on each PRI. This is done

in a gated subset whose parameters are set at initialization time and, of course, should not be located where a target might be expected. By positioning this gate correctly, wake cross section can also be measured.

5.4 High Sensitivity Mode

This is a mode which has been reserved for a special restricted processing application for the DSP in which up to 20 dB of additional sensitivity would be attainable. It is envisioned as an acquisition aid to enable the target to be acquired and tracked before the single pulse signal to noise ratio warrants the use of the mission mode processing. It is equally applicable to satellite detection and tracking as the early detection of re-entry vehicles.

ALCOR, operating at C-band, must designate the target range for this mode will only process three adjacent range cells, to enable range tracking, and the two angle error channels. The method employed is the pulse to pulse coherent integration of the signal in the AP-120B, while performing the actual pulse compression essentially in parallel in the I/O processor, known as the GPIOP. The proper technique for integration is the use of an FFT so that fixed errors in target velocity have no effect on the detection process. Uncertainty in acceleration is permitted up to about

$$\delta \ddot{R} = \frac{2\lambda}{T^2} \qquad T = \text{coherent integration interval}$$

Above this value, a search in acceleration would be required using different quadratic phase compensations across the integration interval.

The software to implement the pulse to pulse coherent integration in the AP-120B is essentially straight forward and poses no risk. For this reason it remains unimplemented. The success of this processing mode depends critically upon whether or not the GPIOP can perform its intended function, and even if so, at what rate. Because of this critical element, the GPIOP software has been written and tested, at least to the extent possible. It will be described in the next section.

The execution time of the GPIOP software has been measured at under 400 μsec thus allowing all of the received energy from a target to be processed at the full 2000 Hz PRF capability of the radar. The maximum integration gain is now limited by either the target coherence time, the maximum track update time, or the available buffer space in the AP's main memory. With the existing processing parameters chosen and memory available, the latter limits the integration to about 100 pulses or 20 dB.

By developing the high risk software, it is felt that the proof of concept for this mode of operation has been achieved and the completion of the software will be a relatively low risk effort.

6. I/O PROCESSOR SOFTWARE

The main I/O processor in the AP-120B is called the GPIOP. In this application there are two active tasks resident. The primary one is to transfer the contents of the aux buffer and track data buffer to the main memory. This is activated by an interrupt from the radar timing system at the end of receive time. A secondary task is the transfer of the range amplitude signal to the display subsystem. This is activated by an interrupt from the AP to the GPIOP after the PP signal has been processed.

The block diagram of the GPIOP software is shown in Figure 7. After the completion of the transfer of the radar data the AP will reset the GPIOP for the next transfer. The GPIOP then waits for the next interrupt from the timing system. When this is received, the entire aux buffer is transferred to main memory as 16 bit integers.

The first word of this buffer is decoded to determine how many radar samples to read from the track data buffer, whether the angle and/or OF channels should be transferred or not, and if the high sensitivity mode is in effect. If it is not, the radar data is transferred to main memory converting from 16 bit integers to AP floating point numbers on the fly. The conversion process limits the transfer rate to 1.5 MHz, thus requiring just over 1 msec to transfer 256 complex samples from 3 channels. In this mode, the arithmetic logic unit in the GPIOP is used strictly for computing

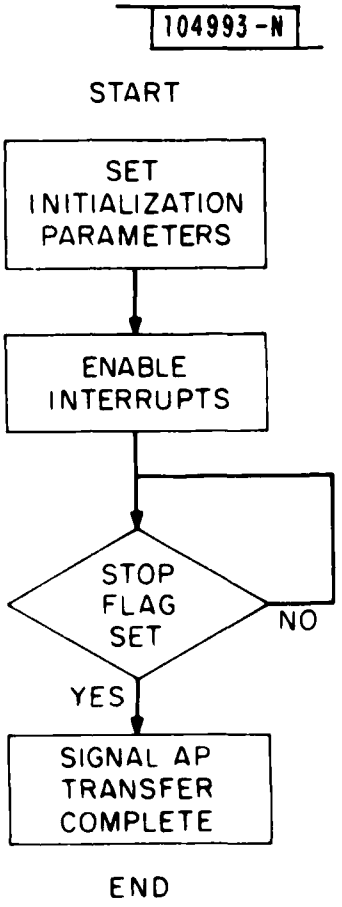


Fig. 7 GPIOP software.

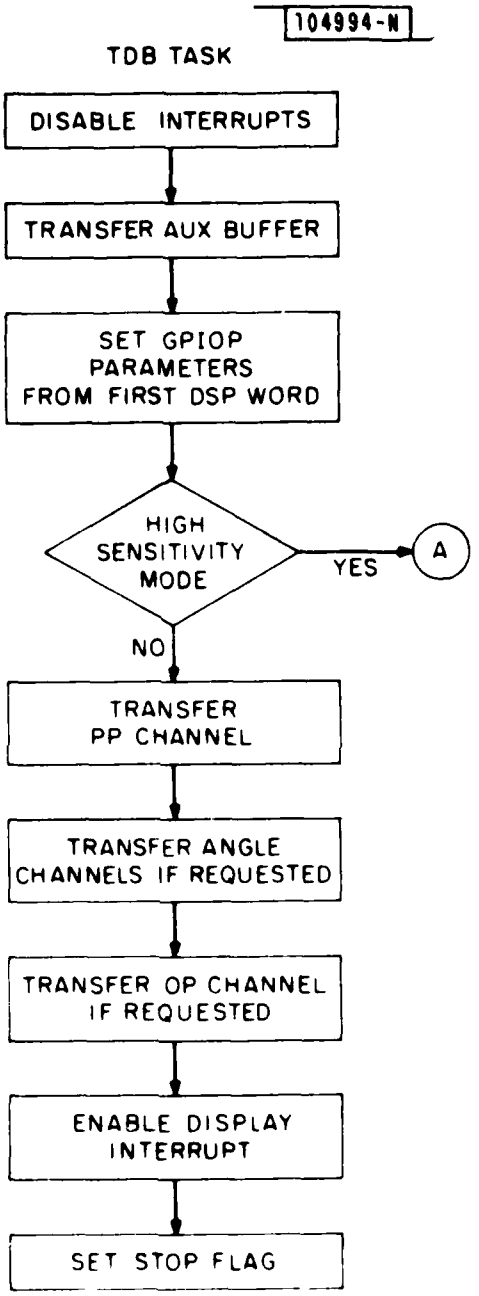


Fig. 7. Continued.

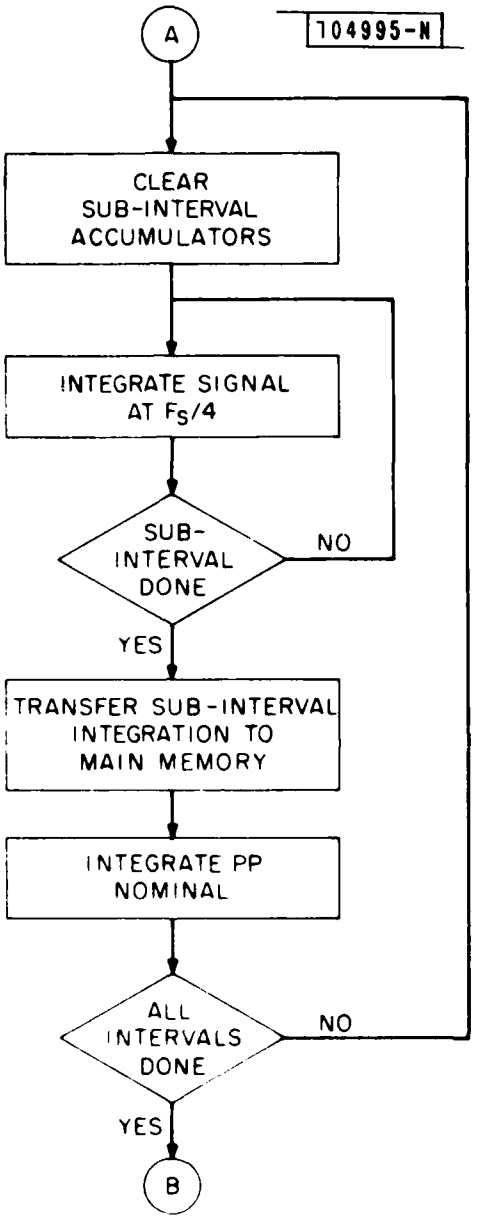


Fig. 7. Continued.

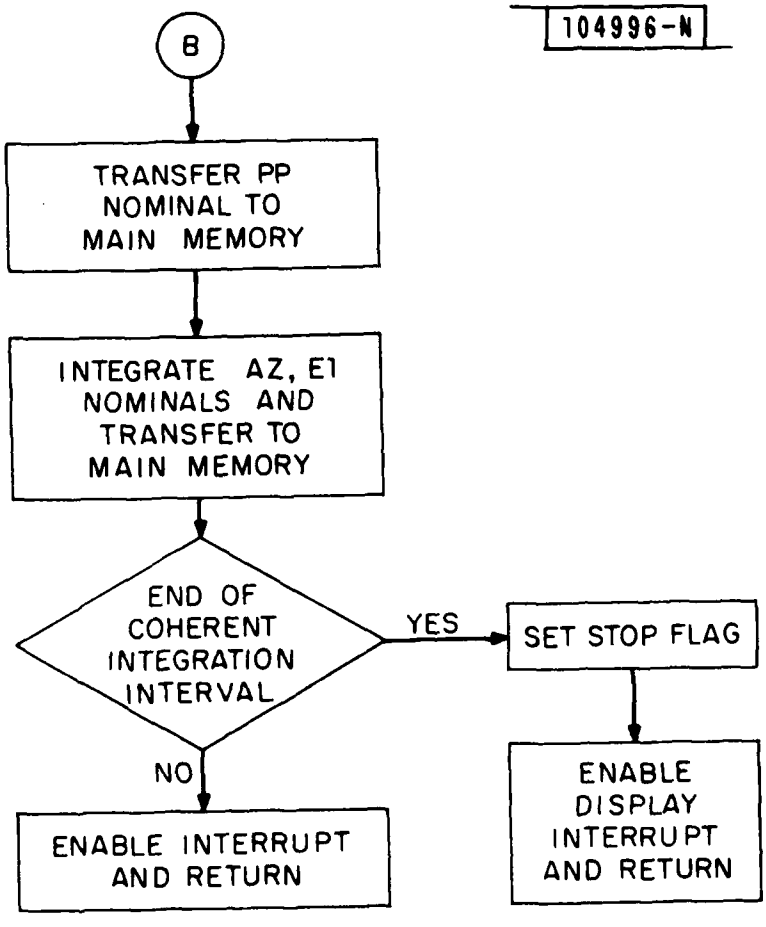


Fig. 7. Continued

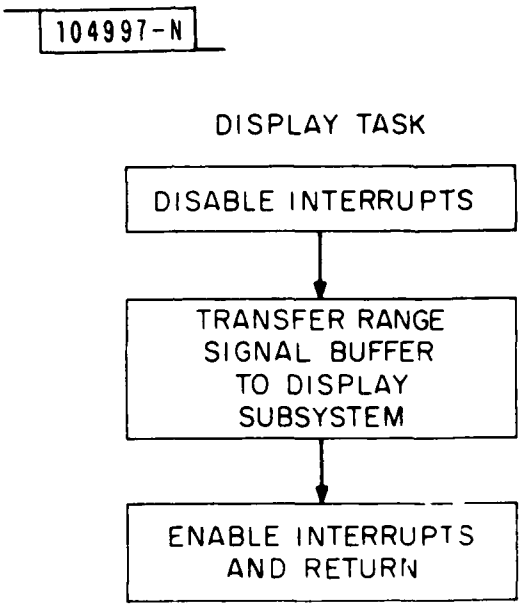


Fig. 7. Continued.

destination addresses in the memory and does not operate on the radar data at all. This is to be contrasted with the high sensitivity mode in which the radar samples are transferred into the 20 bit arithmetic logic unit in the GPIOP where the pulse compression of a single range bin will be performed. In order to allow for range tracking, the range bins on either side of the nominal are partially compressed into a series of subintervals where the elements of each subinterval receive the same phase shift. The phase shift from one interval to the next is applied in the AP itself thus not requiring the GPIOP to have any multiplication capability. In addition to the PP channel, both of the angle channels are compressed in the nominal range bin. The OP channel is not processed in this mode. To eliminate interference from the D.C. offset that will exist, the nominal range is offset from the center of the range window by 1/4, i.e., A/D sampling frequency / 4. In this case the resulting phase shifts that are applied to the signal for the pulse compression are multiples of 90 degrees thus also not requiring any multiplication capability.

Because of the restricted capability of the GPIOP's ALU, this mode, of necessity, does not have the flexibility of the other modes. The number of subintervals in the uncompressed range window is currently set to 16 and there are assumed to be 256 complex samples in each channel. Both of these can only be altered by reassembling the program. With the current parameters, 380 μsec is needed to perform the processing on each PRI, certainly well within the 500 μsec that is available at the maximum radar PRF. How-

ever, if a double buffer is not used in the external hardware, the radar pulse length must be subtracted yielding 450 μsec. This now limits the time available for the display to 70 μsec which would be a 128 point display. A couple of observations can be made concerning this mode: one is that the hardware should be double buffered and although not indicated explicitly above, must be capable of 6 MHz operation. Another is that the display should not be restricted to a minimum of 256 samples.

7. FUNCTIONAL INTERFACE SPECIFICATION

7.1 Aux Buffer and Track Data Buffer

At the end of receive time, an interrupt (INT00*) will be sent to the GPIOP from the recording system. There will be five separate buffers, one each for the aux and the PP, Az, El, and OP channels. For all modes except the high sensitivity mode, the peak read rate will not exceed 3 MHz. When operating in the high sensitivity mode, the PP, Az, and El buffers will be read at a peak rate of 6 MHz. The reading is accomplished by a program in the GPIOP executing an IN instruction which causes INB* to go low for that and only that instruction. Strobes for incrementing external address counters may be formed by gating INB* with the 6 MHz clock from the GPIOP.

To read each buffer, the GPIOP will set the appropriate bit in the device control register according to Table 3. The hardware is expected to initialize the memory address register and have an auto-increment mode.

TABLE 3

DEVICE CONTROL REGISTER BIT ASSIGNMENTS

Device/Channel Assignment	Device Control Register Bit Assignment
TDB reset	19
Aux read	18
PP "	17
Az "	16
El "	15
OP "	14
Display	13

The buffer is then read by consecutive IN instructions. This process is repeated for each buffer that is read and each buffer should be independent of the others. The number of words read from a buffer will be greater than the number of radar samples collected because of the pipelined nature of the GPIOP.

The aux data buffer will contain 16 bit integers, right justified in the AP 38 bit word. The radar data should be 2's complement integers, right justified, sign extended to 20 bits, first the real component and then the imaginary component in two transfers. At the end of the last transfer for each PRI, a pulse 1/6 μsec wide, will be sent out using bit 19 of the device control register.

The AP-120B software follows a mathematical convention for signs in an FFT computation, which requires that the hardware should have the provision for interchanging the L.O.'s in the in-phase and quadrature demodulators. All four channels must be consistent in the phase of the L.O. at the demodulator and what is termed the "real" or "imaginary" channel.

7.2 Display Subsystem

The transfer of data from the GPIOP to the display will take place in the following manner:

- Bit 13 of the device control register in the GPIOP will be set (low true).
- The transfer of N+2 words will commence.
- Word 1: Right justified integer (the second DSP word from the Aux buffer) indicating which display to use.
- Word 2: Right justified integer containing the number of words (N) to be displayed, either 256 or 512.
- Word 3 to N+2: Signal power to be displayed in 38 bit AP floating point format.

At the end of the transfer the device control register will be cleared. The peak transfer rate will not exceed 3 MHz but may be expected to reach this value. In an analagous manner to reading, the transfer is accomplished by executing an OUT instruction in the GPIOP which causes OUTB* to go low for that cycle.

8. TIMING MEASUREMENTS

To support various radar system functions, the DSP was required to cycle in not more than 5 milliseconds when processing with a 256 point FFT in the prime mission mode. In fact, the actual cycle time is just about 3 msec. Table 4 gives an estimated breakdown of this number with the signal processing overhead and the DMA interference being the most unreliable estimates.

TABLE 4

PROCESSING TIME BREAKDOWN

executive level software		0.05 msec
signal processing vector operations		
weighting	2/3 μsec	0.17
FFT		0.97
magnitude	2/3	0.17
range marking	1/3	0.09
angle channel	3 x 2/3	0.51
signal processing overhead (15%)		0.29
DMA interference (input) max		0.64
DMA interference (output) max		0.11
host interface		0.05
TOTAL		3.05 msec

Actual measurements have been made and the results are given in Table 5. This table reflects the various options that are available in mission mode which may be selected via the first two DSP control words in the aux buffer. By processing only the PP channel, for example, a PRF of over 500 Hz can be achieved with a 256 point FFT.

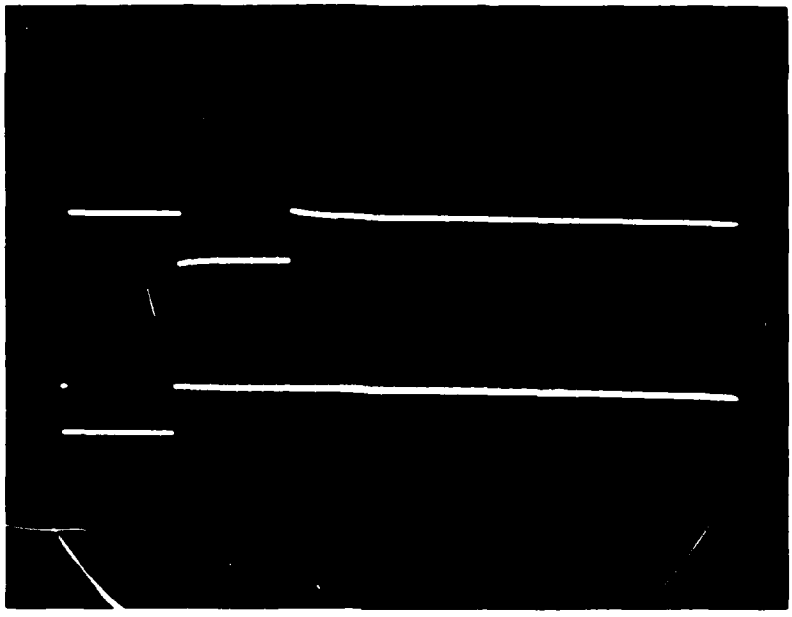
The characteristics of some of the signals that go back out to the hardware from the GPIOP are shown in Figures 8 through 12. These are representative of wide band mission mode. In Figure 8, the scope is triggered by INT00*, the upper trace is DC16* which is low when the Az channel is being read, and the lower trace is DC17* which corresponds to the PP channel. In Figure 9, the upper trace is INB* and the lower is DC17* showing that portion which is being read from the PP channel. Figure 10 is an expanded view showing the detail of INB*. This signal is low for one machine cycle (1/6 μsec) for each word that is being read and runs at 1.5 MHz rate. Contrast this with Figure 11 which show the reading of the aux buffer. Here, no format conversion takes place and the transfer runs at a 3 MHz rate. Seventeen words are shown being read, this includes 16 actual words and 1 for priming the pipeline.

The display is sent out as soon as the magnitude of the PP signal is available, before the target detection or angle channel processing takes place. In Figure 12, the scope is triggered by the timing system interrupt, the

TABLE 5

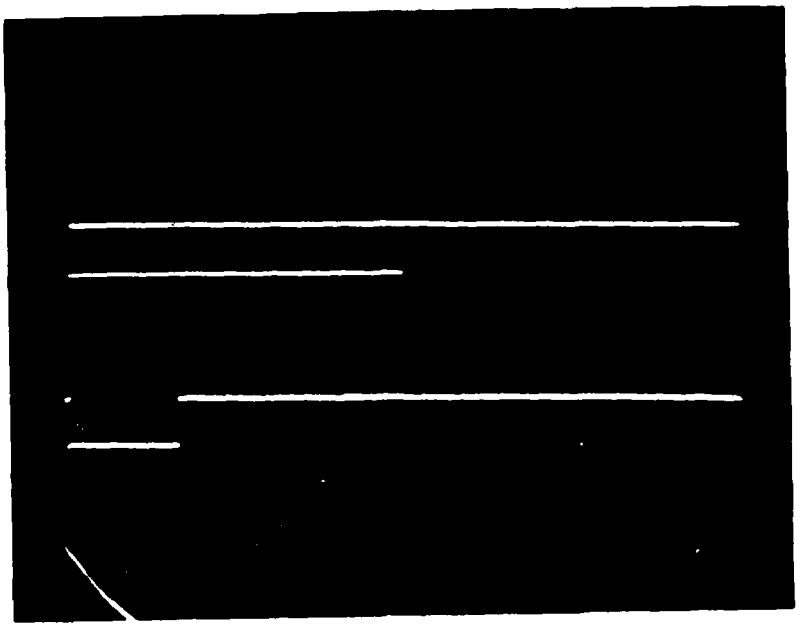
MISSION MODE PROCESSING TIMES

FFT	Angles	OP	Time
256	no	no	1.9 msec
256	no	yes	2.5
256	yes	no	2.9
256	yes	yes	3.3
512	no	no	3.9
512	no	yes	5.1
512	yes	no	5.8
512	yes	yes	6.6



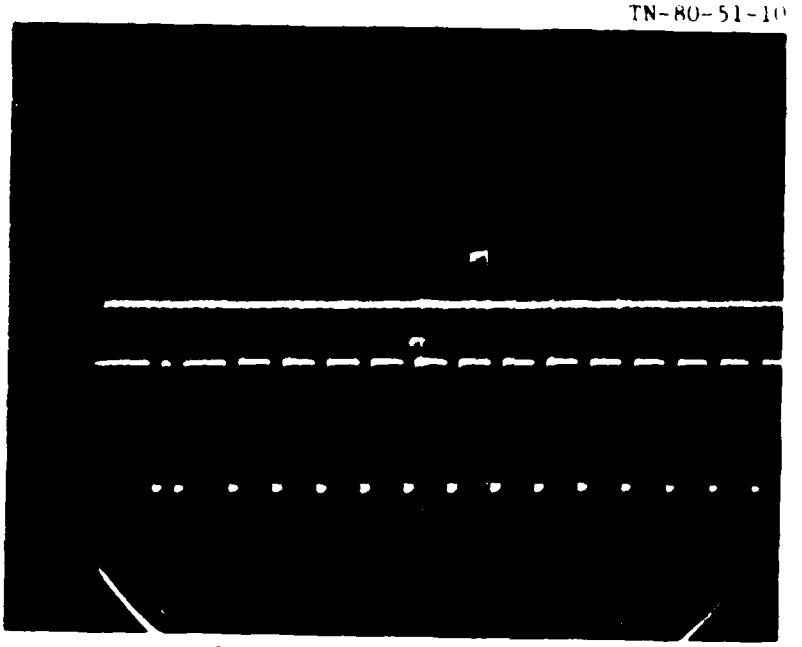
Horizontal: 200 μsec/cm

Fig. 8. Az and PP Channels.



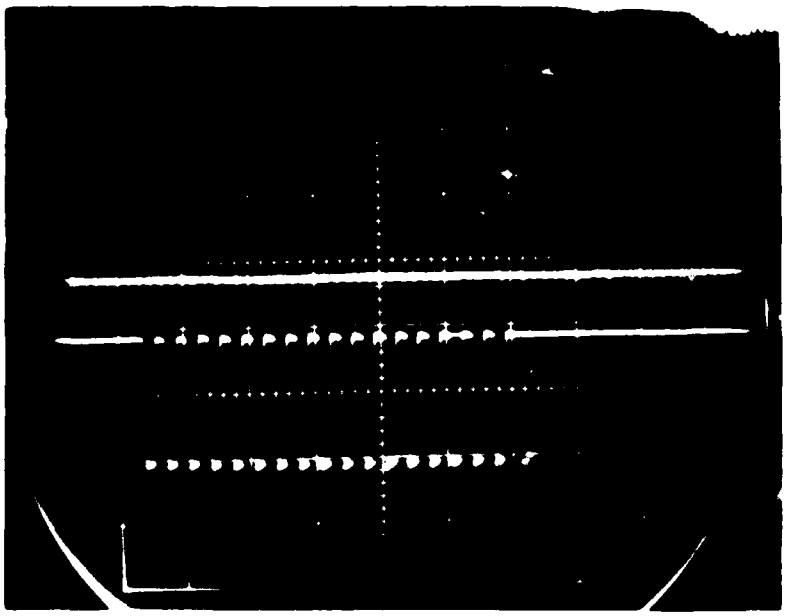
Horizontal: 200 µsec/cm

Fig. 9. INB* and PP Channel.



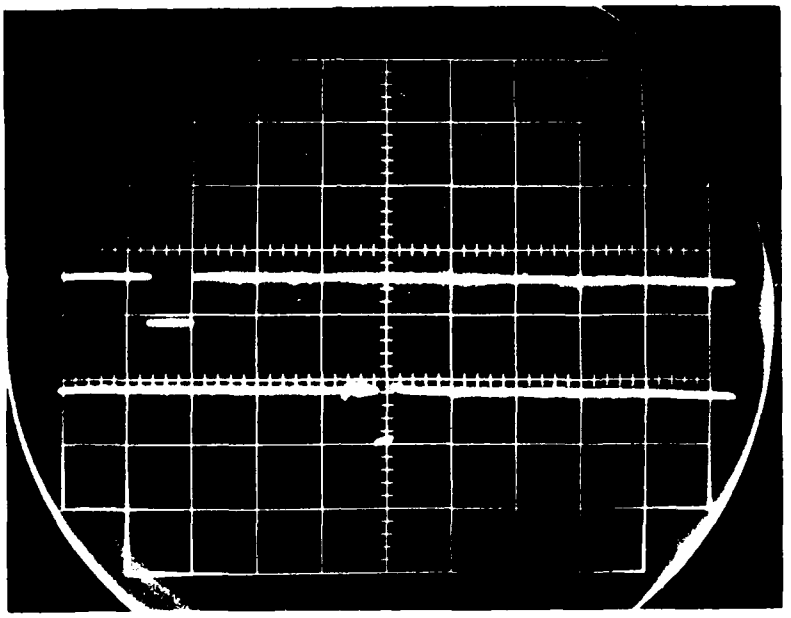
Horizontal: 1 μsec/cm

Fig. 10. INB* and Az Channel.



Horizontal: 1 sec/cm

Fig. 11. Aux Butter and iNB*.



Horizontal: 0.5 msec/cm

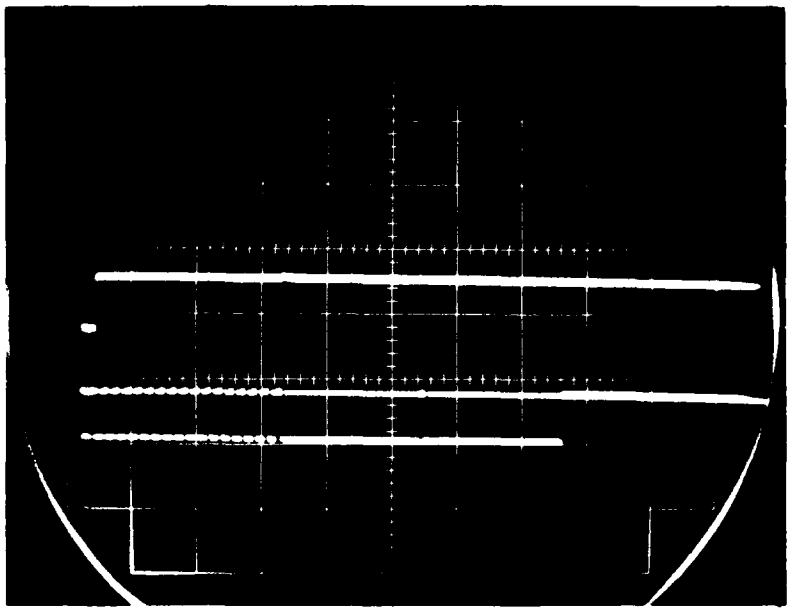
Fig. 12. Display Transfer After Reading Elevation Channel

top trace show the last channel, E1, being read, and the bottom trace shows the display being transferred out 1.4 msec later. This transfer requires 125 μsec to complete. The software in the GPIOP that is performing the transfer is actually capable of running at 3 MHz which would only require about 90 μsec to complete the transfer. The discrepancy is because there is a program executing in the AP at the same time which is competing for memory. The DMA interference is causing the additional 35 μsec.

The signals in the high sensitivity mode are shown in Figures 13-16. In the first of these, the top trace is DC18* for the aux buffer and the bottom is INB*. The different method of reading the PP and angle channels is evident and results from the range tracking requirement. The entire transfer takes less than 400 μsec. Figure 14 shows the PP channel being read. Here can clearly be seen the 16 subintervals into which the range signal is subdivided to enable the partial compression of the pulse in the GPIOP. The angle channel close-up is shown in Figure 15. The lower trace, INB*, is low most of the time indicating an exceptionally high read rate. The peak burst rate is 6 MHz and the average is 4.8 MHz. Figure 16 is a close-up of INB* for the PP channel showing the complicated reading that is being done during the pulse compression.

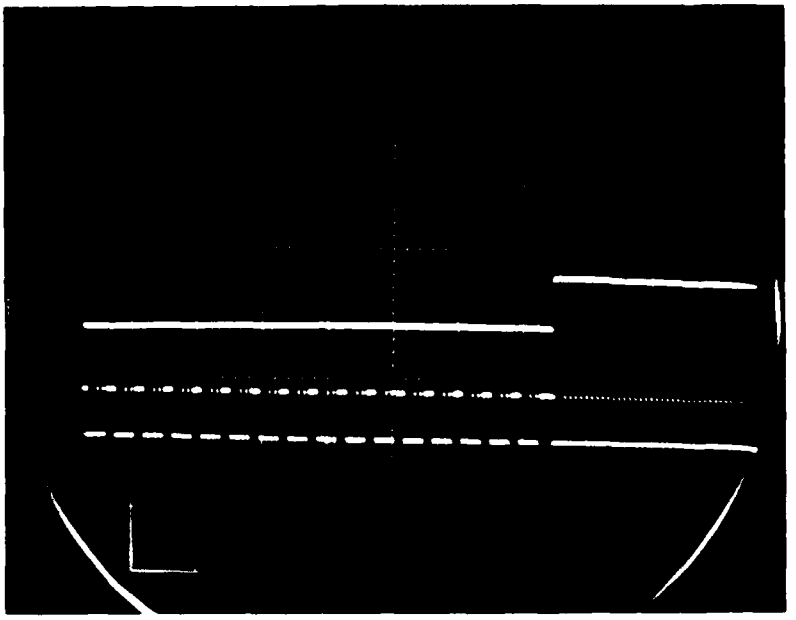
9. MODCOMP INTERFACE

The host computer for the DSP is a Modcomp Classic. This host interface, as supplied, was ill-suited for the real-time application that has been



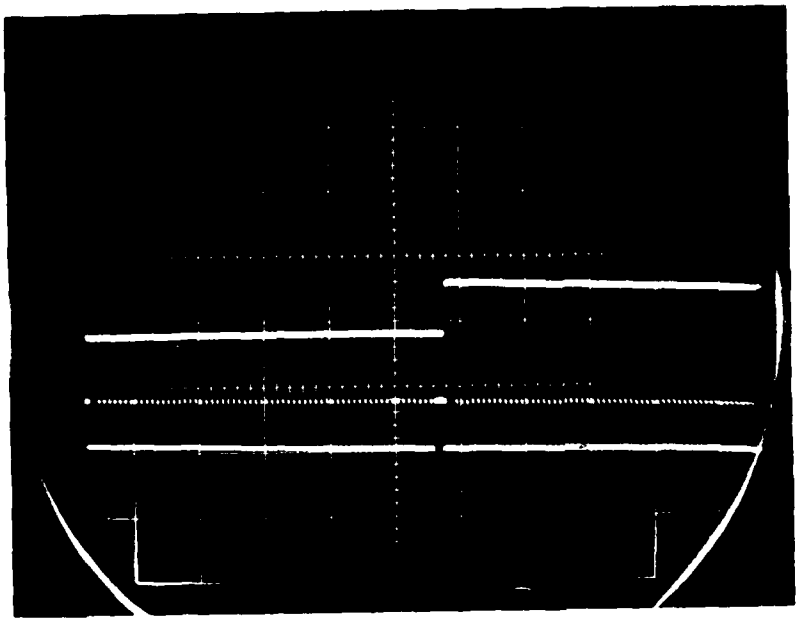
Horizontal: 50 µsec/cm

Fig. 13. Aux Buffer and INB*.



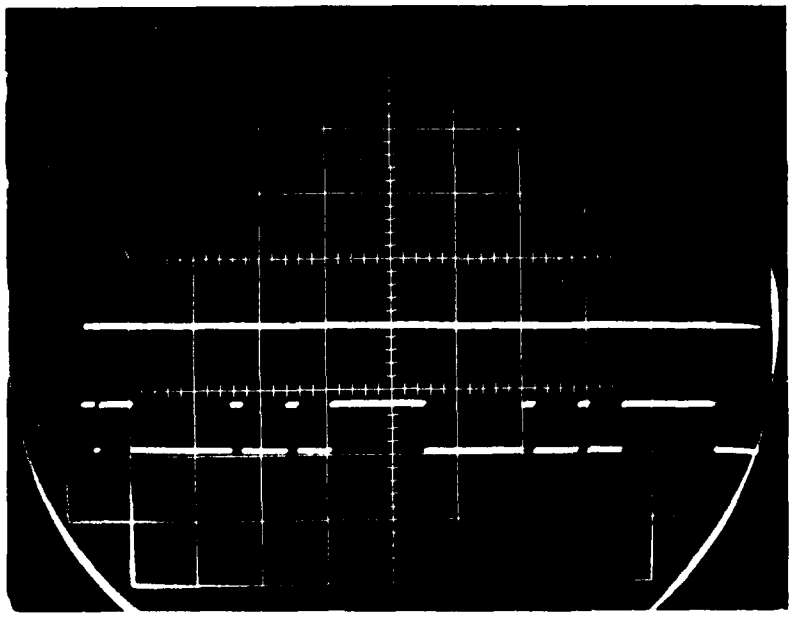
Horizontal: 20 μsec/cm

Fig. 14. Expanded View of PP Channel.



Horizontal: 20 µsec/cm

Fig. 15. Angle Channels.



Horizontal: 2 μsec/cm

Fig. 16. Close-up of PP Channel.

described in this report, from both a hardware and software point of view. At the time of this writing, the philosophy of implementation was still in a state of flux. The description that follows therefore is what is considered the most desirable means of interfacing and is indeed the goal, but with the understanding that the final interface description may differ, especially with respect to chaining.

At the end of a processing sequence, the DSP transfers its output buffer to the Modcomp and then interrupts the Modcomp at the end of the transfer. The format of the buffer is mode dependent but consists of a chained transfer of a small floating point buffer, a fixed point buffer, and another floating point buffer. The integers are always transferred but both of the floating point buffers are optional. Tables 6 and 7 describe these buffers more completely.

The essence of the routine which communicates with the Modcomp is given as a flow diagram in Figure 17. The DMA can only be active upon entry if a problem exists. After the first transfer in the chain is initiated by the AP, a test is made to see if it got started within 5 μsec. If so, the program will wait a reasonable amount of time for the transfer to complete. "Reasonableness" is based on 14 Modcomp words at a 500 KHz transfer rate and can be very well defined. Upon completion, the second transfer of the chain is started and, as before, a test is made to see if it actually did get started within the specified time.

TABLE 6

OUTPUT BUFFER FOR TEST AND CALIBRATION MODES

Fixed Point Buffer

Word	Contents
0	detection flag (-1 = no detection)
1	idle time, LSB = 100 μsec
2	AP status word
3	GP error code
4	AP error code
5	sequence count
6-7	spares
8-23	Track Aux Buffer

Floating Point Buffer

This buffer consists of N1 complex samples from the PP channel, N2 from Az, N3 from El, and N4 from OP. N1, N2, N3, and N4 are given in the fixed point parameter buffer which was loaded at initialization time.

TABLE 7

MISSION MODE OUTPUT BUFFER

Floating Point Buffer

Word	Contents
0	range error
1	spare
2	traverse monopulse
3	elevation monopulse
4	PP target amplitude
5	OP amplitude (if processed)
6	noise level

Fixed Point Buffer

Identical to mode 0.

Floating Point Buffer

If transferred, this buffer contains the amplitudes from the gated subset of the range window. The size of the gate determines the number of words transferred.

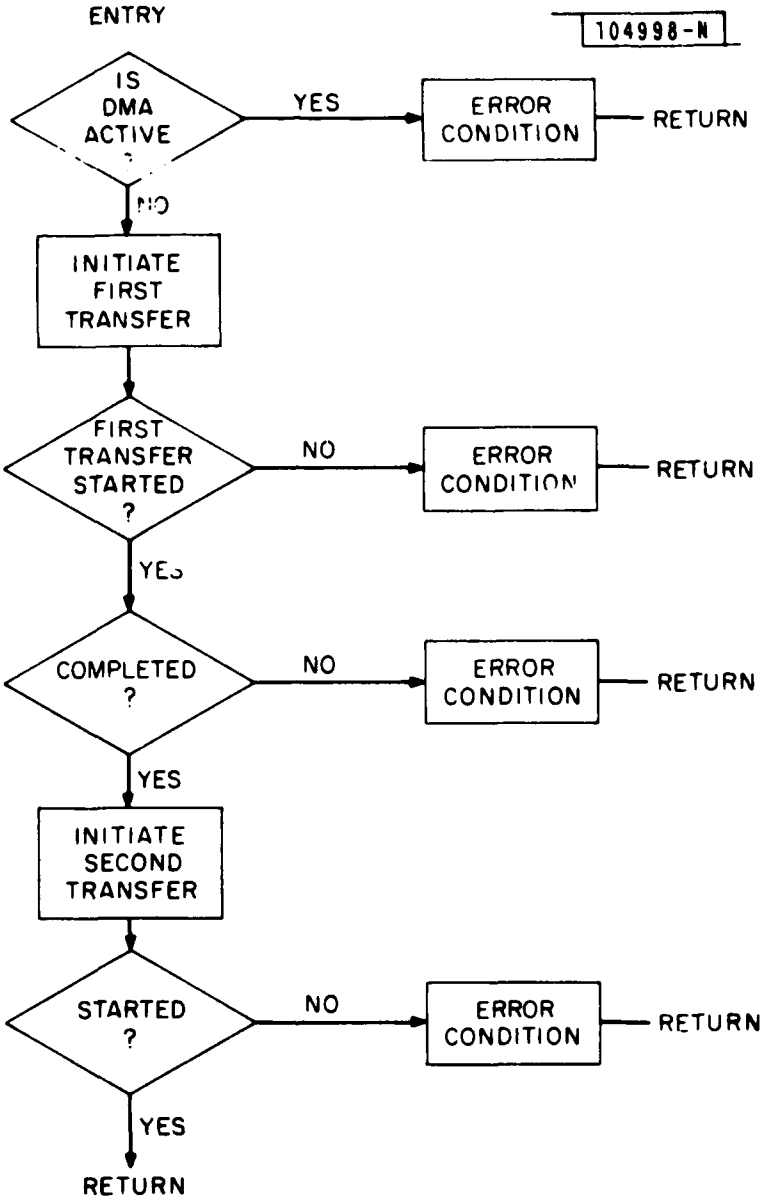


Fig. 17. Host transfer flow chart.

CONCLUSION

The use of an array processor in the digital processing subsystem of this radar has been very successful. It has exceeded, by a comfortable margin, all of the original design specifications and has shown additional capabilities that doubtless will someday become radar requirements. The value of the synchronous architecture to be found in this processor cannot be overemphasized in reducing the level of programming effort required for its use.

The experience gathered in using array processors on two major radar systems is invaluable in specifying certain requirements for the next generation of programmable signal processors. The value of the synchronous architecture in reducing the level of programming effort cannot be overstated. In the author's opinion, it represents at least a factor of two in the simple vector coding and even more in complex routines. It is necessary that the contents of the arithmetic and memory pipelines be saved as previously discussed in Section 4. Several million words of directly addressable memory are required and would mean an expansion of the memory address word to 24 bits. The instruction repertoire of the s-pad section is primitive at best resulting in gross inefficiencies in the use of program memory when parameters are simply being shuffled about prior to

use of the arithmetic elements in the vector mode. Large amounts of data coming in via DMA transfers cause interference in memory accesses resulting in additional execution time. A separate memory accessed by the GPIOP and the AP could eliminate this. Since often, radar processing involves complex operations on complex data, the inclusion of complex arithmetic hardware is a natural means of increasing throughput. The requirement for floating point hardware would never by a negotiable item but the use of a 28 bit mantissa far exceeds any reasonable dynamic range requirement.

GLOSSARY

A/D	analog to digital
AGC	automatic gain control
ALU	arithmetic logic unit
Az	azimuth
DMA	direct memory access
DSP	digital signal processor
El	elevation
FFT	fast Fourier transform
I/O	input/output
L.O.	local oscillator
LRIR	long range imaging radar
Mflops	millions of floating point operations per second
OP	orthogonal polarization
PP	principle polarization
PRF	pulse repetition frequency
PRI	pulse repetition interval
TE	transversal equalizer
TTL	transistor transistor logic

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-A102 535
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle) Real Time Signal Processor for the ALCOR Millimeter Wave Radar	5. TYPE OF REPORT & PERIOD COVERED Technical Note
6. PERFORMING ORG. REPORT NUMBER Technical Note 1980-51	
7. AUTHOR(s) Robert A. Frosch	8. CONTRACT OR GRANT NUMBER(s) F19628-80-C-0002
9. PERFORMING ORGANIZATION NAME AND ADDRESS Lincoln Laboratory, M.I.T. P.O. Box 73 Lexington, MA 02173	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Program Element Nos. 63304A-63308A Project No. 2129
11. CONTROLLING OFFICE NAME AND ADDRESS Ballistic Missile Defense Program Office Department of the Army 5001 Eisenhower Avenue Alexandria, VA 22333	12. REPORT DATE 6 November 1980
	13. NUMBER OF PAGES 66
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Electronic Systems Division Hanscom AFB Bedford, MA 01731	15. SECURITY CLASS. (of this report) Unclassified
	15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

None

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

array	pulse compression	range marking
signal processing	ALCOR	monopulse

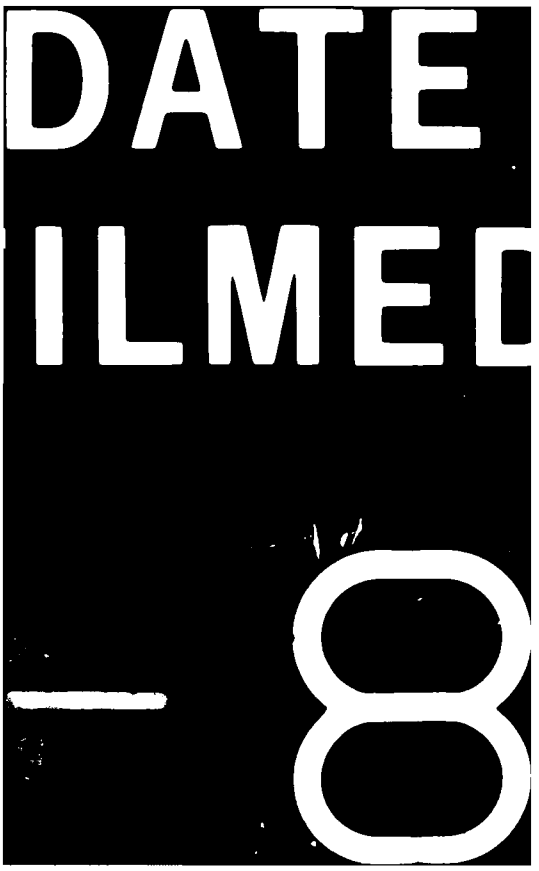
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes the use of an array processor for real time radar signal processing. Pulse compression, range marking, and monopulse error computation are some of the functions that will be performed in the array processor for the millimeter wave ALCOR radar augmentation. Real time software design, processor architecture, and system interfaces are discussed in the report.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)


DATE
ILMED
8